



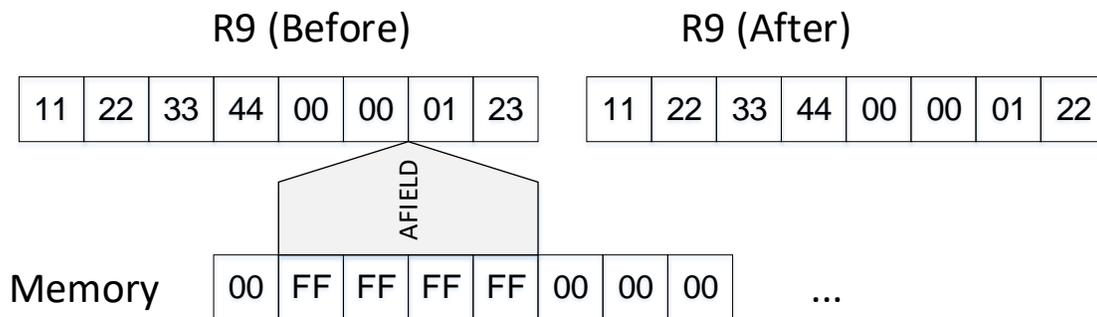
E3	R ₁ X ₂	B ₂ DL ₂	DL ₂ DL ₂	DH ₂ DH ₂	5A
----	-------------------------------	--------------------------------	---------------------------------	---------------------------------	----

The Add instruction performs 2's complement binary addition. Operand 1 is a register containing a fullword integer. Operand 2 specifies a fullword in memory. The fullword in memory is added to the fullword in the register and the result is stored in the register. The fullword in memory is not changed. The leftmost bits (bits 0 – 31) of the target register are unchanged by this operation. Bits 32-63 contain the resulting sum.

Since **AY** has an RXY-a instruction format, the second operand base/displacement address has a displacement size of 20 bits and is treated as a signed binary integer. This supports a displacement range from -524,288 0 to 524,287. This is significantly larger than the 12-bit displacement (which is treated as a non-negative integer) of an add fullword instruction (A) which has a range from 0 – 4095.

Consider the following example,

AY R9, AFIELD



The contents of the fullword “AFIELD”, x'FFFFFFFF' = -1 in decimal, are added to register 9 which contains x'00000123' = 291 in decimal. The sum is decimal 290 = x'00000122' = 290 in decimal and destroys the previous value in R9. The fullword in memory is unchanged by this operation.

Since **AY** is an RXY-a instruction, an index register may be coded as part of operand 2.

AY has a two-byte opcode, E35A. The displacement bytes are not stored in sequence in the object code.



Some Unrelated AYs

R4 = X'00000000FFFFFFFE' -2 IN 2'S COMPLEMENT (bits 32-63)
 R5 = X'0000000000000028' +40 IN 2'S COMPLEMENT (bits 32-63)
 R6 = X'0000000000000004' +4 IN 2'S COMPLEMENT (bits 32-63)

DOG DC F'4'
 CAT DC F'-4'

AY	R4, =F' 20'	R4 = X'0000000000000012' = +18
AY	R5, =F' 20'	R5 = X'000000000000003C' = +60
AY	R6, =F' 20'	R6 = X'0000000000000018' = +24
AY	R6, =F' -5'	R6 = X'00000000FFFFFFF' = -1
AY	R6, CAT	R6 = X'0000000000000000' = 0
AY	R6, DOG	R6 = X'0000000000000008' = +8
AY	R6, DOG (R6)	R6 = X'0000000000000000' INDEXING IS ALLOWED

Tips

1. **AY** can reach further with a 20-bit signed displacement off the base address of a base register than **A** which has a 12-bit non-negative displacement. It can also reach forward or backward from the base address to reference an operand. You might consider using **AY** instead of **A** in many cases. Rather than adding another base register to fix an addressability error, consider using **AY** to help solve your addressability problems.
2. Many instructions have companions in the RXY-a instruction class. RXY-a instructions provide 20-bit signed displacements (range -524,288 to 524,287) compared to 12-bit displacements (range 0 – 4095) of RX instructions. For example, SY is the companion Subtract Fullword operation of S.